Recoverable systems on lines and grids

Alexander Barg Ohad Elishco Ryan Gabrys Eitan Yaakobi

***Abstract*—A storage code is an assignment of symbols to the vertices of a connected graph $G(V,E)$ with the property that the value of each vertex is a function of the values of its neighbors, or more generally, of a certain neighborhood of the vertex in G. Under the name of recoverable systems, a class of storage codes on $\mathbb{Z}$ was recently studied relying on methods from constrained systems and ergodic theory. In this work, we address the question of the maximum capacity of recoverable systems on $\mathbb{Z}$ and $\mathbb{Z}^2$ from a combinatorial perspective. We establish a closed form formula for the capacity of several one- and two-dimensional systems, depending on their recovery set, using connections between storage codes, graphs, anticodes, and difference-avoiding sets.**

I. Introduction

The concept of storage codes on graphs was introduced in [10], [15] as an extension of the notion of locally recoverable codes to systems with restricted connectivity. Let $G(V,E)$ be a connected graph with $|V| = n$ and let $\mathcal{Q}$ be a finite alphabet of size q. A *storage code* $\mathcal{C} \subseteq \mathcal{Q}^n$ is a collection of n-words such that for every $x \in \mathcal{C}$ and every vertex $v \in V$ the value x_v is a function of the values $(x_u, u \in \mathcal{N}(v))$, where $\mathcal{N}(v) := \{u : (v,u) \in E(G)\}$. The main problem concerning storage codes is establishing the largest size of $\mathcal{C}$ for a given class of graphs. It was soon realized that this problem can be equivalently phrased as the smallest rate of symmetric index codes [3], or the largest success probability in (one variant of) guessing games on the graph G [13]; see [1], [4] for a more detailed discussion.

For an integer $n \geqslant 0$ we denote $[n] = \{0, 1, \dots, n-1\}$ with $[0] = \varnothing$, and for a finite set of integers R, we write $n + R := \{n + r \ : \ r \in R\}$. For a sequence $x \in \mathcal{Q}^V$ and for a set of integers R, denote by x_R the restriction of x to the positions in R. The construction problem of storage codes on infinite graphs was recently studied in [7], with a focus on the case of $V = \mathbb{Z}$ and $(i,j) \in E$ if and only if $|i-j| = 1$. The authors of [7] considered codes X formed of bi-infinite sequences $x \in \mathcal{Q}^{\mathbb{Z}}$ with the property that for any $i \in \mathbb{Z}$ the value x_i is found as $f(x_{i+R})$, where $R = \{j : 0 < |j| < l\}$ and $f : \Sigma^{2l} \to \mathcal{Q}$ is a deterministic function, independent of i, and they coined the term *recoverable systems* for such codes. Paper [7] additionally assumed that the system X is shift invariant, i.e., if $x \in X$ then also a left shift $Tx \in X$ (T acts by shifting all the symbols in x one place to the left). This assumption enabled them to rely on methods from constrained systems to estimate the growth rate of the set of allowable sequences, or the capacity of recoverable systems.

In this paper we study recoverable systems on $\mathbb{Z}$ and $\mathbb{Z}^2$ (lines and grids) that are not necessarily shift invariant, and that can rely on recovery functions that depend on the location a of the entry x_a of the word. This will enable us to use methods from storage/index coding to compute or bound the capacity of such systems.

II. Preliminaries

The following definition formalizes the concept of storage codes discussed above for the case of $V = \mathbb{Z}^d, d \geqslant 1$.

Definition 1. *Let $R \subset \mathbb{Z}\backslash\{0\}$ be a finite subset of integers ordered in a natural way. An R-*recoverable system *$X = X_R$ on $\mathbb{Z}$ is a set of bi-infinite sequences $x \in \mathcal{Q}^{\mathbb{Z}}$ such that for any $i \in \mathbb{Z}$ there is a function $f_i : \mathcal{Q}^{|R|} \to \mathcal{Q}$ such that $x_i = f_i(x_{i+R})$ for any $x \in X$.*

*More generally, let $R \subset \mathbb{Z}^d\backslash\{0\}$ be a finite subset together with some ordering. An R-*recoverable system *$X = X_R$ on $\mathbb{Z}^d$ is a set of letter assignments (d-dimensional words) x such that for any vertex $v \in \mathbb{Z}^d$ there is a function $f_v : \mathcal{Q}^{|R|} \to \mathcal{Q}$ such that $x_v = f_v(x_{v+R})$ for any $x \in X$. Here $v + R = \{v + z : z \in R\}$ is a translation of v by the recovery neighborhood (set) R.*

This definition is close to the definition of storage codes [10] which also allows the dependency of the recovery function on $v \in V$ and also (implicitly) assumes an ordering of the vertices in V. Here we wish to note an important point: by defining the recovery set R we effectively introduce an edge (possibly, a directed one) between v and every vertex in $v + R$. Thus, we speak of R-recoverable systems on $\mathbb{Z}^d$ with edges defined by R, and denote the graphs that emerge in this way by $\mathbb{Z}^d_R$ (or G_R for general G). If $d = 1$, we omit it from the notation.

Example 2. *Take $V = \mathbb{Z}$ and $R = \{-1, 1\}$. Assume that $x_{2i+1} = x_{2i}$ for all i. Then we have $f_i(x_{i-1}, x_{i+1}) = x_{i+(-1)^i}$, that is, $f_i(x_{i-1}, x_{i+1}) = x_{i-1}$ or x_{i+1} depending on whether i is odd or even.* □

Let $X = X_R$ be an R-recoverable system. For $n \geqslant 0$, denote by $B_n(X)$ the restriction of the words in X to the set $[n]$, i.e., $B_n(X) = \{x_{[n]} : x \in X\}$. The *capacity* of X is defined as

$$\mathsf{cap}(X) := \limsup_{n\to\infty} \frac{1}{n} \log_q |B_n(X)|. \tag{1}$$

Formally we should write cap_q, but below the value of q does not play a role, and is omitted.

Alexander Barg is with ISR/Dept. of ECE, University of Maryland, College Park, MD, USA and also with IITP, Russian Academy of Sciences, Moscow, Russia. Email: abarg@umd.edu. His research was supported by NSF grant CCF2104489 and NSF-BSF grant CCF2110113.

Ohad Elishco is with Ben-Gurion University of the Negev, Israel, Email: ohadeli@bgu.ac.il. His research was supported by NSF-BSF grant CCF2020762.

Ryan Gabrys is with the University of California at San Diego, CA, USA. Email: ryan.gabrys@gmail.com.

Eitan Yaakobi is with the Technion - Israel Institute of Technology, Haifa, Israel. Email: yaakobi@cs.technion.ac.il.

978-1-6654-2159-1/22/$31.00 ©2022 IEEE

Given a recovery set $R \subset \mathbb{Z}\backslash\{0\}$, we are interested in its largest attainable capacity

$$\mathsf{cap}(R) := \sup_{X_R \subseteq \mathcal{Q}^{\mathbb{Z}}} \mathsf{cap}(X_R) \tag{2}$$

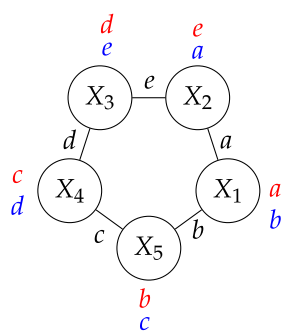
of R-recoverable systems on $\mathbb{Z}$. These definitions are naturally extended for two and higher dimensions.

Example 3. *Consider the R-recoverable system from Example 2. Since every odd symbol is uniquely determined by the two adjacent symbols in even positions, we have $B_n(X) \leqslant q^{\lceil n/2 \rceil}$, and since every symbol can occur in the even positions, we have $B_n(X) \geqslant q^{\lfloor n/2 \rfloor}$. Thus, $\mathsf{cap}(X) = \frac{1}{2}$.* □

A. Storage codes for finite graphs

We briefly mention the known results for storage codes on finite graphs as defined in the opening paragraph of the paper. Let G be a graph with n vertices and let $\mathcal{R}_q(\mathcal{C}) := \frac{1}{n}\log_q|\mathcal{C}|$ be the rate of a q-ary code $\mathcal{C}$ on G. In this case the recovery set of a vertex v is $\mathcal{N}(v)$, and it depends on v, so we denote the largest rate of a storage code on G by $\mathsf{cap}(G)$.

There are several ways of constructing storage codes with large rate. The most well-known one is the edge covering construction: given a (d-regular) graph, place a q-ary symbol on every edge and assign each vertex a d-vector of symbols written on the edges incident to it (again we assume an ordering of the edges). The size of the code is $(q^d)^{n/2}$, resulting in the rate value $1/2$ irrespective of the value of q. Here is an example with $q = d = 2$:



This method extends to the case when every vertex is incident to the same number of cliques. For instance, if this number is one, then the graph can be partitioned into k-cliques. To construct a code, we put a single parity on every clique and distribute the symbols of the parity to the vertices that form it, resulting in rate $\mathcal{R} = (k-1)/k$. A further extension, known as *clique covering* [3], states that

$$\mathsf{cap}(G) \geqslant 1 - \alpha(G)/n, \tag{3}$$

where $\alpha(G)$ is the smallest size of a clique covering in G. Among other general constructions, we mention the *matching construction*, which yields

$$\mathsf{cap}(G) \geqslant M(G)/n,$$

where $M(G)$ is the size of the largest matching in G.

Turning to upper bounds, we note a result in [3], Theorem 3 (also [10], Lemma 9), which states that

$$\mathsf{cap}(G) \leqslant 1 - \gamma(G)/n, \tag{4}$$

where $\gamma(G)$ is the independence number of G, i.e., the size of the largest independent set of vertices (in other words, the size of the smallest vertex cover of G).

The definition of storage codes can be generalized by permitting a recovery "neighborhood" other than the set $\mathcal{N}(v)$, such as the set R discussed in Def. 1. Effectively, this changes the connectivity of the graph, so while we keep the same set of vertices V, the edges are now drawn according to where the failed vertex collects the data for its recovery (these new graphs were denoted by G_R above, assuming that R is the same for every vertex). Since R does not have to be symmetric, the graph G_R generally is directed (as is often the case in index coding [2], [3]). In this case, bound (4) affords a generalization, which we proceed to describe.

For a subset of vertices $U \subseteq V$ in a graph $G = (V, E)$, denote by $G(U)$ the induced subgraph. A directed graph with no directed cycles is called a *directed acyclic graph* (DAG). It is known that a graph is a DAG if and only if it can be topologically ordered [9, p. 258], i.e., there is a numbering of the vertices such that the tail of every arc is smaller than the head. A set of vertices $S \subseteq V$ in a graph G is called a *DAG set* if the graph $G(S)$ is a DAG. With this preparation, the following *MAIS bound* is true, see [3], Theorem 3, or [2], Sec. 5.1.

Theorem 4. *Let $G = G(V, E)$ be a graph and let $\delta(G)$ be the size of the largest DAG set in it. Then*

$$\mathsf{cap}(G) \leqslant 1 - \delta(G)/n. \tag{5}$$

The proof follows by noticing that if we puncture the nodes in a DAG set S, their values can be recovered following the topological ordering of the vertices. Other upper bounds on the rate of storage codes are found in [11] for the symmetric case and in [2] for the general case.

B. Recoverable systems on $\mathbb{Z}$

Bounds and constructions for finite graphs can be extended to infinite graphs such as $\mathbb{Z}^d$. In this section we state a few easy results that exemplify the above methods for the case of $d = 1$, noting that analogous claims can be stated in higher dimensions.

Let $R \subset \mathbb{Z}\backslash\{0\}$ be a finite set and consider R-recoverable systems on the infinite graph $\mathbb{Z}_R$. For all $n > 0$ let us consider the subgraph $\mathbb{Z}_{R,n} := \mathbb{Z}_R \cap [n]$. To argue about the capacity $\mathsf{cap}(R)$ we first find $\mathsf{cap}(\mathbb{Z}_{R,n})$ and argue that these quantities are related. This follows from the observation that we can disregard the boundary effects because the proportion of points whose recovery regions do not fit in $[n]$ vanishes as n increases. Thus, we can place constant values on these points with a negligible effect on the rate of the code.

Lemma 5. *Let $R \subset \mathbb{Z}\backslash\{0\}$ be a finite set and $(c_n)_{n=1}^{\infty}$ an infinite sequence of real numbers such that $\mathsf{cap}(\mathbb{Z}_{R,n}) \leqslant c_n$ for all $n > 1$. Then, $\mathsf{cap}(R) \leqslant \limsup_{n\to\infty} c_n$.*

Proof: For all $n > 0$, let A_n be the set of vertices in $\mathbb{Z}_{R,n}$ in which their neighborhood in the graph $\mathbb{Z}_R$ does not fully belong to the graph $\mathbb{Z}_{R,n}$, i.e., $A_n = \{m \in [n] : m + R \nsubseteq [n]\}$. Since $|R|$ is a constant independent of n, starting from some value of n we have $|A_n| \leqslant a$ for some absolute constant a.

Let X be an R-recoverable system. For a given vector $z = (x_i)_{i \in A_n} \in \mathcal{Q}^{|A_n|}$, let $B_n^z(X)$ be the set of words in $B_n(X)$ which match the values of z over the positions in A_n. The code $B_n^z(X)$ is a storage code over the graph $\mathbb{Z}_{R,n}$ and hence

$$n^{-1} \cdot \log_q |B_n^z(X)| \leqslant \mathsf{cap}(\mathbb{Z}_{R,n}) \leqslant c_n.$$

Since the inequality holds for all $z \in \mathcal{Q}^{|A_n|}$ and $|A_n| \leqslant a$, we conclude that

$$n^{-1} \cdot \log_q |B_n(X)| \leqslant c_n + a/n,$$

which verifies the lemma's statement. ■

According to Lemma 5, it is possible to calculate the capacity of several recovery sets R.

Proposition 6. *Let l, r be positive integers, let $R = \{1 + [r]\} \cup \{-1 - [l]\}$, and let $m = \min\{l, r\}$. Then,*

$$\mathsf{cap}(R) = \frac{m}{m+1}.$$

Proof: Assume without loss of generality that $r \leqslant l$, so $m = r$, and let $\mathbb{Z}_R$ be the graph describing the R-recoverable system.

The upper bound follows by noting that for all $n > 1$ the set of vertices $\{k(r+1) : k \in \mathbb{Z}\} \cap [n]$ is a DAG set in the graph $\mathbb{Z}_{R,n}$ and thus

$$\mathsf{cap}(\mathbb{Z}_{R,n}) \leqslant 1 - \frac{\delta(\mathbb{Z}_{R,n})}{n} \leqslant 1 - \frac{\lceil \frac{n}{r+1} \rceil}{n} \leqslant \frac{r}{r+1}.$$

Together with Lemma 5 this implies that $\mathsf{cap}(R) \leqslant r/(r+1)$.

To show the reverse inequality we use the clique partition construction (3). Start with partitioning $\mathbb{Z}$ into segments of length $r+1$, setting $\mathbb{Z} = \bigcup_{k \in \mathbb{Z}} k \cdot [r+1]$. We aim to construct a code in which every vertex in the segment can be recovered from the other vertices in it. In other words, $\mathbb{Z}_R$ is a disjoint union of $(r+1)$-cliques, and the construction described above before (3) yields a code of rate $r/(r+1)$. ■

In the case of *shift invariant* recoverable systems the upper bound of this proposition was derived in [7]; however, the authors of [7] stopped short of finding a matching construction under this assumption. The challenge in the shift invariant case is to find a recoverable system with the *same* recovery function for all symbols.

The next proposition follows by similar arguments.

Proposition 7. *Let l, r be positive integers and let $R = \{-l, r\}$. Then for any $q \geqslant 2$*

$$\mathsf{cap}(R) = \frac{\gcd(l, r)}{l + r}.$$

Example 8. *To give an example, consider a system X_R with $R = \{-6, 4\}$, so that $\gcd(4, 6) = 2$. Now put a codeword of the length-5 repetition code on each of the mutually disjoint sets of consecutive symbols in the even positions and in the odd positions. For instance, we can take those sets to be $\{10k + 2j\}$ and $\{10k + 2j + 1\}$ with $0 \leqslant j \leqslant 4, k \in \mathbb{Z}$. Then clearly for every $i \in \mathbb{Z}$, the symbol x_i is a part of the codeword of the repetition code that includes either position $i - 6$ or position $i + 4$, and we can recover x_i by accessing one of those positions as appropriate.* □

Let $B \subset \mathbb{Z}_+$ be a finite or infinite set. We call a set $A \in \mathbb{Z}$ *B-avoiding* if its difference set $|A - A|$ is disjoint from B, i.e., $|a_1 - a_2| \notin B$ for all $a_1, a_2 \in A$. Let a_n be the size of the largest set $A_n \subseteq [n]$ that is B-avoiding and define $\mathcal{R}(B) = \limsup_{n \to \infty} a_n/n$. This quantity has been extensively studied in the literature, initially with B being the set of all whole squares [14] and later values of other polynomials; see, e.g., [12].

Proposition 9. *Let $0 < r_1 < r_2 < \cdots < r_s$ and $0 < l_1 < l_2 < \cdots < l_t$ be positive integers and $R = \{-l_t, \ldots, -l_2, -l_1, r_1, r_2, \ldots, r_s\}$. Then,*

$$\mathsf{cap}(R) \leqslant 1 - \max\{\mathcal{R}(\{l_1, \ldots, l_t\}), \mathcal{R}(\{r_1, \ldots, r_s\})\}.$$

Proof: Assume without loss of generality that $\mathcal{R}(\{l_1, \ldots, l_t\}) \leqslant \mathcal{R}(\{r_1, \ldots, r_s\})$ and let A_n be the largest subset of $[n]$ that is $\{r_1, r_2, \ldots, r_s\}$-avoiding. Then, A_n is a DAG set in the graph $\mathbb{Z}_{R,n}$ and thus $\mathsf{cap}(\mathbb{Z}_{R,n}) \leqslant 1 - |A_n|/n$. The statement follows from Lemma 5. ■

Consider for example the set $B_m = \{1, 2, 4, \ldots, 2^m\}$ for $m \geqslant 1$. Then, $\mathcal{R}(B_m) = 1/3$ (achieved by the B_m-avoiding set $\{3j : j \geqslant 0\} \cap [n]$ for all n). Then, according to Proposition 9, it is possible to derive that for all $t, s \geqslant 1$, and $R = \{-2^t, \ldots, -4, -2, -1, 1, 2, 4, \ldots, 2^s\}$, $\mathsf{cap}(R) \leqslant 1 - 1/3$. Equality holds in this case since by Proposition 6, $\mathsf{cap}(R) \geqslant \mathsf{cap}(\{-2, -1, 1, 2\}) = 2/3$.

Next we discuss how to draw connections between the capacity of one-dimensional recovery sets and the capacity of recovery sets in two (and more generally, higher) dimensions.

Let G be a graph with a vertex set $[n]$. For a subset $S \subset [n]$ and $a, 0 < a < n$ we denote $a + S := \{(a + s)(\bmod n) : s \in S\}$. A DAG set $S \subseteq [n]$ in G is called *invariant* if for all $0 < a < n$, the set $a + S$ is also a DAG set. We further say that G satisfies the MAIS bound (5) *invariantly* if it has an invariant DAG set whose size satisfies (5) with equality. Finally, we call a recovery set R *invariant*, if for all sufficiently large n, the graph $\mathbb{Z}_{R,n}$ satisfies the MAIS bound invariantly.

Proposition 10. *Let $R \subset \mathbb{Z} \backslash \{0\}$ be a finite invariant set and let $R^2 = ([0] \times [R]) \cup (R \times [0]) \subset \mathbb{Z}^2 \backslash \{0\}$. Then, $\mathsf{cap}(R^2) = \mathsf{cap}(R)$.*

Proof: For all sufficiently large n, assume that S_n is an invariant DAG set in $\mathbb{Z}_{R,n}$. Then, the set

$$S_n^2 = \bigcup_{0 \leqslant i \leqslant n-1} ([i, i] + [0] \times S_n)$$

is a DAG set in the graph $\mathbb{Z}_{R^2,n}$. Thus, $\mathsf{cap}(\mathbb{Z}_{R^2,n}) \leqslant 1 - |S_n|/n = \mathsf{cap}(Z_{R,n})$. ■

III. Storage capacity and codes in graphs

The independent set bound (4) can be interpreted in terms of packings of graphs which we explain using the language of coding theory. Codes in graphs form a classical topic in combinatorial coding; see, e.g., [6, Ch.2], [5].

Given a finite graph $G(V, E)$, we define the graphical distance $\rho(u, v)$ as the length of the shortest path in G between

u and v. A code in G is a subset $\mathcal{C} \subset V$, and it is said to have minimum distance d if $\rho(u,v) \geqslant d$ for all pairs of distinct vertices $u, v \in \mathcal{C}$, or, in other words, if for any two distinct code vertices $u, v \in \mathcal{C}$ the balls $B_r(u)$ and $B_r(v)$ of radius $r = \lfloor (d-1)/2 \rfloor$ are disjoint. A code $\mathcal{C} \subset V$ is called an r-covering code if $\bigcup_{v \in \mathcal{C}} B_r(v) = V$. We denote by $C(G,r)$ the smallest size of an r-covering code in the graph G.

Below we use properties of codes in graphs that depend essentially on symmetry, i.e., for any pair of vertices u, v, if v is a part of the recovery region of u (denoted $R(u)$) then u is contained in $R(v)$. We will moreover assume that the graph G is distance-regular [5]. For instance, an $n \times n$ region of $\mathbb{Z}^2$ is distance-transitive and therefore distance-regular except for the vertices near the boundaries. Since the proportion of these vertices out of n^2 becomes negligible for large n (recall again that R is finite), the remarks before Lemma 5 and the lemma itself extend to any number of dimensions $d \geqslant 2$. Below we use these arguments for the two-dimensional case.

Suppose that we are given a finite set $V, |V| = n$ and a subset $R(v) \subset V \backslash \{v\}$ that serves the recovery region for the vertex v. In this section the recovery regions will be given by balls of a given radius in the metric ρ, the same for every vertex $v \in V$. We construct a graph $G_r = G(V,E)$ by connecting each vertex v with all the vertices u such that $1 \leqslant \rho(v,u) \leqslant r$. For instance, in the next section, V will be an $n \times n$ region of $\mathbb{Z}^2$ and $R(v) = \mathcal{B}_r(v) \backslash \{v\}$, where $\mathcal{B}_r(\cdot)$ is a ball of radius r in some metric on $\mathbb{Z}^2$ (we focus on l_1 and l_∞ distances).

Denote by $A(G; r+1)$ the size of the largest code in G with minimum distance $\geqslant r+1$. Any such code is an independent set in G_r, and thus by (4)

$$\mathsf{cap}(G_r) \leqslant 1 - \frac{1}{n}\gamma(G_r) = 1 - \frac{1}{n}A(G; r+1). \qquad (6)$$

Furthermore, every ball of radius $\lfloor r/2 \rfloor$ in G is a clique in the graph G_r, and thus any $\lfloor r/2 \rfloor$-covering code in G can form a clique covering of the graph G_r. Let $\alpha(G_r)$ be the number of cliques in the smallest covering. We obtain that

$$\mathsf{cap}(G_r) \geqslant 1 - \frac{1}{n}\alpha(G_r) \geqslant 1 - \frac{1}{n} C\left(G; \left\lfloor \frac{r}{2} \right\rfloor\right).$$

Denoting by $B_G(r)$ the volume of the ball of radius r in G (because of the regularity assumption, it does not depend on the center), we can use the sphere packing bound to claim that $A(G; r+1) \leqslant \frac{n}{B_G(\lfloor (r-1)/2 \rfloor)}$. If r is even and there exists a perfect $r/2$-error-correcting code then $A(G; r+1) = C(G; r/2) = \frac{n}{B_G(\lfloor (r-1)/2 \rfloor)}$ and so

$$\mathsf{cap}(G_r) = 1 - \frac{1}{n}A(G; r+1) = 1 - \frac{1}{B_G(\lfloor (r-1)/2 \rfloor)}.$$

A subset of vertices in G with bounded *maximum* distance is also called an *anticode*. For instance, a ball of radius τ is an anticode with diameter 2τ. It is known that in many cases the largest size of an anticode of even diameter D is achieved by a ball of radius $D/2$. For odd values of D the largest anticode is usually constructed by taking a union of two balls of radius $(D-1)/2$ whose centers are adjacent in G. A more general version of the sphere packing bound is Delsarte's *code-anticode bound* [6, Thm.3.9] which claims that if $\mathcal{C}$ is a code with minimum distance $r+1$ and $\mathcal{D}$ is an anticode of diameter r, then $|\mathcal{C}||\mathcal{D}| \leqslant |V|$. A code that satisfies this bound with equality is called *diameter perfect*. From (6), for diameter perfect codes,

$$\mathsf{cap}(G_r) \leqslant 1 - \frac{1}{D_G(r)}, \qquad (7)$$

where $D_G(r)$ is the largest size of an anticode of diameter r in the graph G. If there exists a perfect covering of G with anticodes (i.e., every vertex is contained in exactly one anticode), then (7) holds with equality because of (3).

Let us summarize this discussion as follows.

Proposition 11. *If there exists a perfect covering of the graph G_r with anticodes of diameter r and size $D_G(r)$ then*

$$\mathsf{cap}(G_r) = 1 - \frac{1}{D_G(r)}.$$

A. Graphs over the Two-Dimensional Grid

The results in the previous section enable us to study the capacity of graphs that represent a storage network over the two-dimensional grid. We start with recovery sets formed by radius-r balls under the l_1 and l_∞ metrics. For $v = (i_1, j_1), u = (i_2, j_2) \in \mathbb{Z}^2$,

$$d_1\big((i_1, j_1), (i_2, j_2)\big) = |i_1 - i_2| + |j_1 - j_2|,$$
$$d_\infty\big((i_1, j_1), (i_2, j_2)\big) = \max\{|i_1 - i_2|, |j_1 - j_2|\}.$$

The recovery set R_1 under the metric d_1 is sometimes called the *Lee sphere* of radius r, and the recovery set R_∞ under the metric d_∞ is a $(2r+1) \times (2r+1)$ square (both taken without the point $(0,0)$). For a given value of r, we denote the graphs G_{R_1}, G_{R_∞} by G_r^1, G_r^∞, respectively. According to Proposition 11, finding the capacity of the system amounts to constructing a perfect covering of the graph. With this in mind, let us denote the size of a maximum anticode of diameter r in the Lee sphere by $D_1(r)$. It is known [8] that $D_1(r) = (r+1)^2/2$ for odd r and $D_1(r) = r^2/2 + r + 1$ for even r (in the latter case, this anticode is a ball of radius $r/2$). Thus, we obtain the following result.

Proposition 12. *For all $r \geqslant 1$ it holds that*

$$\mathsf{cap}(G_r^1) = 1 - \frac{1}{D_1(r)},$$
$$\mathsf{cap}(G_r^\infty) = 1 - \frac{1}{(r+1)^2}.$$

Proof Sketch: For the graph G_r^∞, anticodes of diameter r are simply squares of size $(r+1) \times (r+1)$, so $D_{G_r^\infty}(r) = (r+1)^2$. The squares tile the graph, giving a perfect covering, and thus

$$\mathsf{cap}(G_r^\infty) = 1 - \frac{1}{(r+1)^2}.$$

The result for the G_r^1 metric is obtained from perfect tiling which exists for maximal anticodes in the Lee metric [8]. ■

Now suppose that the recovery set is not symmetric, for instance, a direct product of two non-symmetric segments. The previous proposition yields the following result.

Corollary 13. *Let l, r, b, a be positive integers and consider the recovery set $R = [-l, r] \times [-b, a]$ where $0 \leqslant r < l$ and $0 \leqslant b < a$. Let X_R be a two-dimensional R-recoverable system on $\mathbb{Z}_2$. Then*

$$\mathsf{cap}(G_R) = 1 - \frac{1}{(r+1)(b+1)}. \tag{8}$$

Proof Sketch: To show that the left-hand side in (8) is less than the right-hand side we simply observe that

$$S = \{(i(r+1), j(b+1)) : i, j \geqslant 0\}$$

is a DAG set in the graph G_R.

For the lower bound, let

$$S_1 = \left\{(i,j) \in \mathbb{Z}^2 \;:\; i \in [r], j \in -[b]\right\}.$$

For $x, y \geqslant 0$, the subset of nodes $(x(r+1), y(b+1)) + S_1$ forms a clique. Conclude by (3). ∎

Another straightforward way to construct two-dimensional systems is to use the *axial product* of two one-dimensional systems. As before, for positive integers l, r, b, a let $R_1 = [-l, r]$ and $R_2 = [-b, a]$. Fix a finite alphabet $\mathcal{Q}$ and consider one-dimensional recoverable systems $Y_1 = Y_{R_1}$ and $Y_2 = Y_{R_2}$. The *axial product* of Y_1 and Y_2 is a two-dimensional system $X = Y_1 \times Y_2$ over $\mathcal{Q}$ with the recovery set $(R_1 \times [0]) \cup ([0] \times R_2)$. In words, in the system X the symbol x_i is a function of the l symbols to its left, r symbols to its right, a symbols above, and b symbols below it. The axial product construction is different from the direct product $[-l, r] \times [-b, a]$ in that it results in a cross-shaped rather than a rectangular recovery region.

Theorem 14. *For given positive integers l, r, b, a, let $R_1 = [-l, r]$ and $R_2 = [-b, a]$, and let $X = Y_1 \times Y_2$ be the axial product of the one-dimensional systems $Y_1 = Y_{R_1}$ and $Y_2 = Y_{R_2}$. Then the capacity*

$$\mathsf{cap}(X) = \frac{t}{t+1}$$

where $t = \max\{\min\{l, r\}, \min\{a, b\}\}$. Moreover, a system X that attains this value can be obtained from the one dimensional R-recoverable system with $R = \{j : 0 < |j| < t\}$.

This theorem shows that for some recoverable systems, the 2D capacity is not increased from the 1D one: the maximally sized system is obtained by stacking independent 1D systems in the rows (or the columns). While the one-dimensional components in the axial product provide an obvious lower bound on the capacity of the 2D system, it is not clear whether the equality holds in all cases. We could not find examples of axial products with higher capacity by relying on both dimensions, and we leave this question as an open problem.

B. The Finite Case

As noted above, the capacity bounds proved in Sec.III apply for finite distance-regular graphs. Previously we employed them to study recoverable systems on $\mathbb{Z}^2$ by looking at $n \times n$ squares and increasing n to find the capacity for various recovery sets. Here we consider the graph given by the $n \times n$ square grid itself. To sidestep the boundary effects, we identify the opposite sides and consider a *discrete torus*, i.e., a graph $\mathcal{T}_n$ on the vertex set $V = [n] \times [n]$ with an edge between (i_1, j_1) and (i_2, j_2) whenever $(i_1 - i_2, j_1 - j_2)$ equals one of $(1,0), (0,1), (-1,0), (0,-1)$ modulo n.

Consider storage codes on $\mathcal{T}_n$ based on the recovery sets given by the entire rows and columns (circles) on the torus. In other words, we take the recovery set in the form $R := (0,*) \cup (*,0) \backslash (0,0)$ where the unspecified coordinates are allowed to vary over the entire set $\mathbb{Z}/(n)$. The graph $(\mathcal{T}_n)_R$ representing the system is obtained from $\mathcal{T}_n$ by connecting all pairs of vertices whose coordinates are identical in either the first or the second position.

Theorem 15. *For $n \geqslant 3$ the storage capacity of the discrete torus $G := (\mathcal{T}_n)_R$ is*

$$\mathsf{cap}(G) = 1 - \frac{1}{n}. \tag{9}$$

Proof: Notice first that for any two vertices (i_1, j_1), (i_2, j_2), there is an edge between (i_1, j_1) and (i_2, j_2) if and only if $d_H((i_1, j_1), (i_2, j_2)) = 1$, where d_H denotes the Hamming distance. Placing a parity constraint on every row of $\mathcal{T}_n$ yields a code of rate $1 - 1/n$, proving a lower bound in (9). In order to show that it is also an upper bound, we recall that from (7) it suffices to show that $D_G(1) = n$. In words, we want to show that the largest anticode of diameter 1 in the graph G under the Hamming metric is of size n, which is immediate.

IV. Conclusion

In this paper we studied the capacity of recoverable systems on $\mathbb{Z}$ and $\mathbb{Z}^2$. Earlier results on this problem [7] relied on the assumption of shift-invariant systems. Lifting this restriction enables one to connect it to the line of research on storage codes for finite graphs. Relying on the known results for the finite case, we found capacity of several examples of recoverable systems. We also pointed out that in the case of distance-regular graphs one can employ Delsarte's code-anticode bound to find the capacity of recoverable sets defined by metric balls under the l_1 and l_∞ distance on the grid. Finally, we established a link of the capacity problem to questions in additive combinatorics related to difference-avoiding sets, and found capacity values for some special recovery sets.

There are numerous open problems that we leave for future research. To point out some of them, one may ask what is the maximum capacity of a two dimensional system when only a subset of neighbors can be used for recovery. Next, it appears that the capacity of an axial product is always attained by stacking 1D systems, although proving this has been elusive. If not true, then what are the conditions under which it is the case? Among other questions: How does the fact that the order of the neighbors is known affect the capacity of the system? And finally, is it possible to characterize the number of recovery functions needed to obtain the maximum capacity?

References

[1] F. Arbabjolfaei and Y. Kim, "Three stories on a two-sided coin: Index coding, locally recoverable distributed storage, and guessing games on graphs," in *2015 53rd Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, 2015, pp. 843–850.

[2] F. Arbabjolfaei and Y.-H. Kim, “Fundamentals of index coding,” *Foundations and Trends in Comm. Inf. Theory*, vol. 14, no. 3-4, pp. 163–346, 2018.

[3] Z. Bar-Yossef, Y. Birk, T. S. Jayram, and T. Kol, “Index coding with side information,” *IEEE Transactions on Information Theory*, vol. 57, no. 3, pp. 1479–1494, 2011.

[4] A. Barg and G. Zémor, “High-rate storage codes on triangle-free graphs,” arXiv:2110.02378.

[5] A. Brouwer, A. Cohen, and A. Neumaier, *Distance-Regular Graphs*. Berlin e.a.: Springer, 1989.

[6] P. Delsarte, “An algebraic approach to the association schemes of coding theory,” *Philips Res. Repts. Suppl.*, no. 10, 1973.

[7] O. Elishco and A. Barg, “Recoverable systems,” arXiv:2010.00589.

[8] T. Etzion, “Product constructions for perfect Lee codes,” *IEEE Transactions on Information Theory*, vol. 57, no. 11, pp. 7473–7481, 2011.

[9] D. E. Knuth, *The Art of Computer Programming. Vol. 1: Fundamental Algorithms*, 2nd ed. Reading, MA: Addison-Wesley Pub. Co., 1973.

[10] A. Mazumdar, “Storage capacity of repairable networks,” *IEEE Transactions on Information Theory*, vol. 61, no. 11, pp. 5810–5821, 2015.

[11] A. Mazumdar, A. McGregor, and S. Vorotnikova, “Storage capacity as an information-theoretic vertex cover and the index coding rate,” *IEEE Trans. Inform. Theory*, vol. 65, no. 9, pp. 5580–5591, 2019.

[12] A. Rice, “A maximal extension of the best-known bounds for the Furstenberg-Sárközy theorem,” *Acta Arith.*, vol. 187, no. 1, pp. 1–41, 2019.

[13] S. Riis, “Information flows, graphs and their guessing numbers,” *Electron. J. Combin.*, vol. 14, no. 1, p. 17pp., 2007.

[14] A. Sárkŏzy, “On difference sets of sequences of integers. I,” *Acta Mathematica Hungarica*, vol. 31, no. 1-2, pp. 125 – 149, 1978. [Online]. Available: https://akjournals.com/view/journals/10473/31/1-2/article-p125.xml

[15] K. Shanmugam and A. G. Dimakis, “Bounding multiple unicasts through index coding and locally repairable codes,” in *2014 IEEE International Symposium on Information Theory*, 2014, pp. 296–300.